

IN THE CLAIMS

Please amend the claims to the following.

1. (Currently Amended) An apparatus comprising:
 - a trigger-response mechanism that includes at least one bank of user-programmable registers to identify a user-defined trigger event; and
 - thread switch handler logic coupled to the trigger-response mechanism to perform a light-weight thread switch of a first instruction pointer from a first thread with a second instruction pointer of ~~to~~ a second thread responsive to the user-defined trigger event occurring during execution of the first thread.
2. (Previously Amended) The apparatus of claim 1, wherein the user-defined trigger event includes a synchronous user-defined trigger event.

1 3. (Previously Amended) The apparatus of claim 1, wherein the user-defined trigger
2 event includes an asynchronous user-defined trigger event.

1 4. (Currently Amended) The apparatus of claim 1, wherein the thread switch handler
2 logic to perform a light-weight thread switch from a first instruction pointer of the
3 first thread to a second instruction pointer of the second thread comprises: saving
4 a first instruction pointer address for the first thread before replacing the first
5 instruction pointer address with ~~setting~~ a second instruction pointer address for
6 the second thread.

1 5. (Previously Amended) The apparatus of claim 4, further comprising: a task queue
2 to receive the first instruction pointer address.

1 6. (Original) The apparatus of claim 5, wherein: the task queue further comprises a
2 memory location.

1 7. (Original) The apparatus of claim 5, wherein: the task queue further comprises a
2 register.

1 8. (Original) The apparatus of claim 1, further comprising: a plurality of event
2 counters coupled to the trigger-response mechanism, wherein each event counter
3 is to detect an atomic processor event.

1 9. (Previously Amended) The apparatus of claim 8, wherein the user-defined trigger
2 event includes an asynchronous trigger event based on one or more of the atomic
3 processor events.

1 10. (Currently Amended) The apparatus of claim 1, wherein the thread switch handler
2 logic is to perform the light-weight thread switch from a first instruction pointer
3 of the first thread to a second instruction pointer of the second thread
4 transparently to an Operating System (OS) and without OS intervention.

1 11. (Cancelled) .

1 12. (Cancelled) .

1 13. (Previously Amended) The apparatus of claim 1, further comprising:
2 one or more user-programmable control registers coupled to the thread switch handler
3 logic;
4 the value of the one or more control registers to indicate a weight of the light-weight
5 thread switch as only an instruction pointer ~~from the first thread to the second thread.~~

1 14. (Currently Amended) A system comprising:
2 a memory to hold an instruction; and
3 a processor coupled to the memory, the processor including raw event detection logic to
4 detect at least one raw event, a user-addressable modifiable register to specify a user-defined
5 trigger event based on the at least one raw event, and a switch handler to invoke a helper
6 thread responsive to the occurrence of the user-defined trigger event.

1 15. (Previously Amended) The system of claim 14, wherein: the instruction includes
2 a marking instruction, when executed, to specify the user-defined trigger event in
3 the user-addressable register.

1 16. (Previously Amended) The system of claim 14, wherein: the instruction is a
2 trigger instruction; and raw event detection logic is to detect an opcode of the
3 trigger instruction when the trigger instruction reaches an execution phase of an
4 execution pipeline.

1 17. (Previously Amended) The system of claim 14, wherein: the processor further
2 includes a user-addressable control register to specify a weight of a context to be
3 saved responsive to invoking the helper thread.

1 18. (Currently Amended) The system of claim 14, wherein: the switch handler is
2 further to maintain minimal context information for a current thread before
3 invoking the helper thread, wherein the minimal context information includes a
4 context weight less than or equal to a full context weight by at least a weight
5 reduced by general register values, of excluding traditional context information.

1 19. (Cancelled)

1 20. (Original) The system of claim 18, wherein the minimal thread context
2 information comprises an instruction pointer address value.

1 21. (Currently Amended) A method comprising:
2 in response to detecting a user-specified trigger condition;
3 suspending execution of a first thread on a threaded processor; and
4 utilizing hardware to switch an amount of context information of the first thread with the
5 amount of context information of a second thread without operating system
6 intervention, wherein the amount of context information has a first weight that is
7 user-defined in a user-addressable control register.

1 22. (Original) The method of claim 21, wherein:
2 detecting a user-specified trigger condition further comprises determining that a trigger
3 instruction has been encountered.

1 23. (Original) The method of claim 21, wherein:
2 detecting a user-specified trigger condition further comprises determining that an
3 asynchronous condition specified in a marking instruction has been encountered.

1 24. (Previously Amended) The method of claim 21, wherein: the first weight includes
2 only an instruction pointer address.

1 25. (Currently Amended) The method of claim 21, further comprising:
2 determining that the first thread should be resumed;
3 restoring the ~~minimal~~-context information having the first weight for the first thread; and
4 resuming execution of the first thread without operating system intervention.

1 26. (Original) The method of claim 21, wherein detecting a user-specified trigger
2 condition further comprises:
3 receiving a marker instruction that specifies the trigger condition; and
4 monitoring a plurality of atomic event indicators to detect the trigger condition.

1 27. (Original) The method of claim 21, wherein detecting a user-specified trigger
2 condition further comprises: generating an asynchronous response to indicate that
3 the second thread should be invoked.

1 28. (Previously Added) A processor comprising:
2 event detection logic to detect a raw event;
3 user-programmable event logic coupled to the event detection logic to indicate a user-
4 defined trigger event, the user-defined trigger event to be based on at least the
5 raw event;
6 user-programmable context control logic to specify a weight of a context to be saved;
7 and
8 thread switch logic coupled to the user-programmable event logic and context control
9 logic, the thread switch logic, in response to the user-defined trigger event being
10 detected, to save a portion of a first context based on the weight of a context to
11 be saved that is to be specified in the user-programmable context control logic
12 and to spawn a helper thread without operating system intervention.

1 29. (Previously Amended) The processor of claim 28, wherein the user-
2 programmable event logic includes at least a user-programmable event register,
3 and wherein the user-defined trigger event is to be programmed in the user-
4 programmable event register in response to execution of a user marking
5 instruction.

1 30. (Currently Amended) The processor of claim 29 [[28]], further comprising trigger
2 response logic coupled to the user-programmable event logic and the event
3 detection logic to detect the user-defined trigger event based on at least the raw
4 event, wherein the trigger response logic is to monitor for the user-defined trigger
5 event for a predetermined timeout period after execution of the user-marking
6 instruction.